

Challenges and Techniques for Implementing Lane Margining

Gopi Krishnamurthy

PCIe® IP Architect

Cadence Design Systems

Copyright © 2017 PCI-SIG® - All Rights Reserved

Disclaimer

<u>Presentation Disclaimer</u>: All opinions, judgments, recommendations, etc. that are presented herein are the opinions of the presenter of the material and do not necessarily reflect the opinions of the PCI-SIG®.

Copyright © 2017 PCI-SIG® - All Rights Reserved

Agenda

- **Review Lane Margining Spec**
- **Review PIPE 4.4.1 Spec**
- **Lane Margining S/W Flow**
- **Example LM Plot**
- **Error Scenarios**
- **Summary**

Copyright © 2017 PCI-SIG® - All Rights Reserved

Why Lane Margining (LM)

- **With PCIe® Link at 16GT/s or higher speeds performance degradation can occur due to**
 - Reflections in the channel
 - Channels loss due to various components in the channel
 - Cross talk resulting in lower noise margins
 - Non optimal jitter tolerance. etc.
- **Above parameters could vary**
 - Mainly due to changes in Voltage, Temp and Humidity
- **It is possible to solve above issues with various design techniques**
 - Important aspect is to identify these problems

Copyright © 2017 PCI-SIG® - All Rights Reserved

Why Lane Margining (LM)

- **No standard way to test and identify the faulty link in a complex system**
 - A data center with hundreds of servers, and each server having many line cards
 - “Live” Debugging of a link may be difficult. Because “Probes” are difficult or some times not possible to be added on to a system
- **CEM (Card Electro Mechanical) testing is expensive**
 - Receiver could behave differently in test mode (Rx Compliance, Loopback) versus “Live” or “mission mode”
 - Introduction of PCIe retimers pose additional challenges
- **No way to know how much margin exists**
 - In a system on all the RX’s “Live”
- **Solution**
 - Lane Margining (LM)

Copyright © 2017 PCI-SIG® - All Rights Reserved

What is LM?

- **Standardized method of margining during L0 through S/W control**
 - No test equipment or NDAs required
 - S/W is used to measure eye and check for margin
- **Allow re-use of existing hardware to support margining**
 - Many PHYs have this capability today, but they are non-standard or proprietary mechanisms
- **Lane Margining mandatory for 16GT/s or higher speeds. Has following information**
 - Time Margining (Left/Right)
 - Voltage Margining is Optional (Up/Down)
- **LM is defined for testing all RX's**
 - EP, RC and any re-timers, so that each point-to-point link can be evaluated.
 - Retimers would need new logic to support Ctrl SKP OS

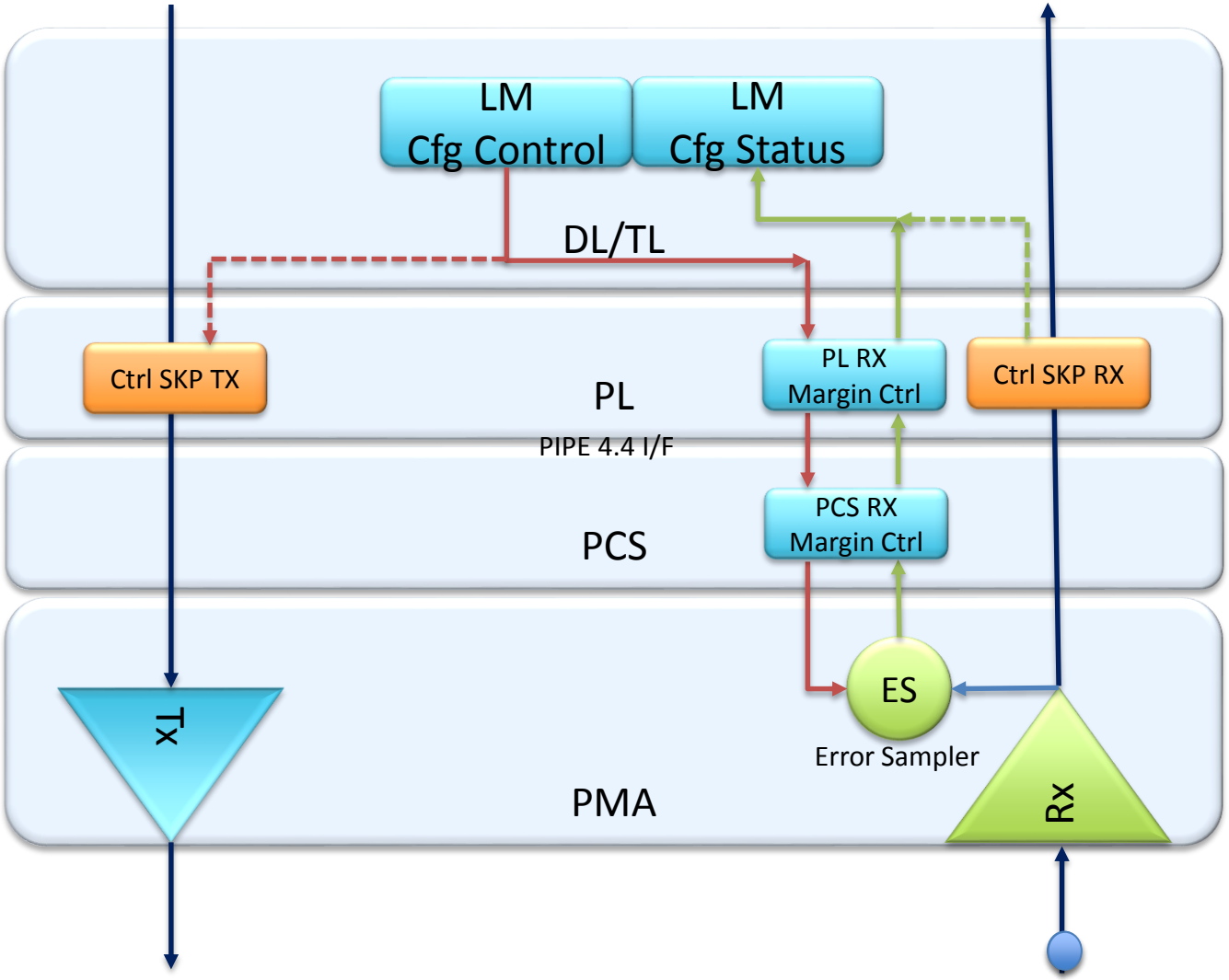
Copyright © 2017 PCI-SIG® - All Rights Reserved

Application of LM

- **Lane Margining could be used for the following purposes**
 - Platform qualification across range of PVT/Components/Systems
 - Debugging in a system without adding extra probes
 - Run-time diagnostics, to measure health of system or system's
 - Checking margin after hot-plug, etc.

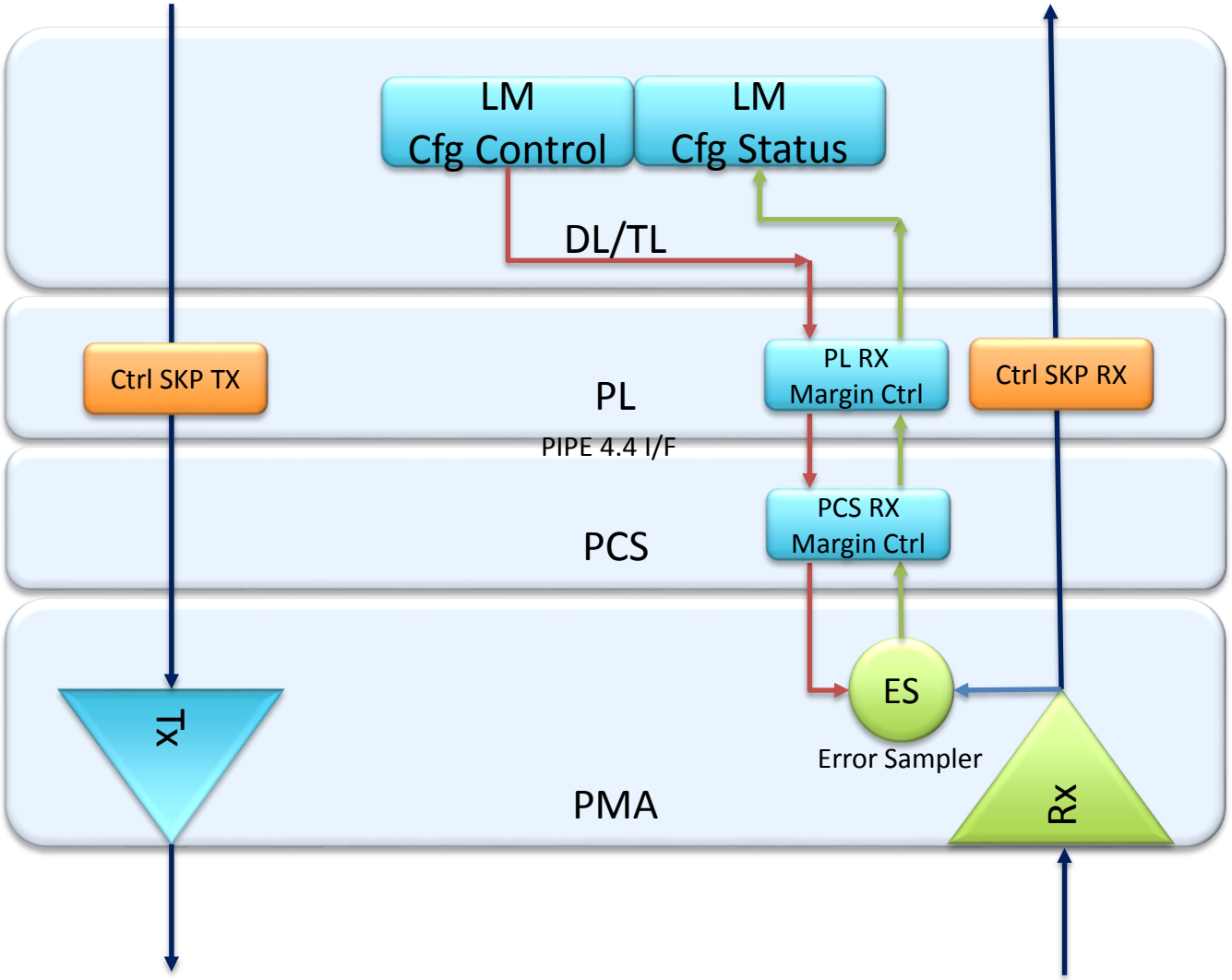
Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Blocks DSP



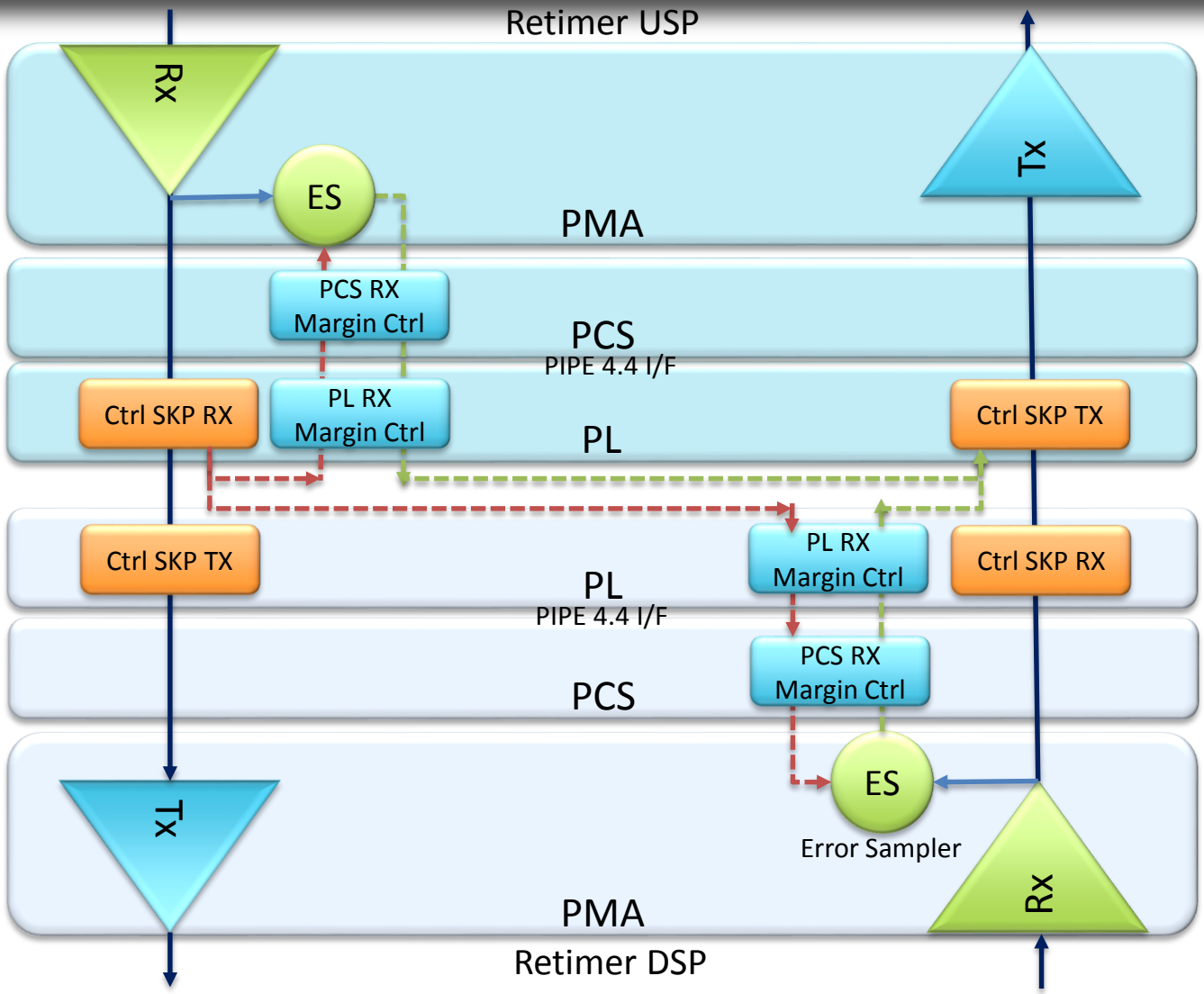
Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Blocks USP



Copyright © 2017 PCI-SIG® - All Rights Reserved

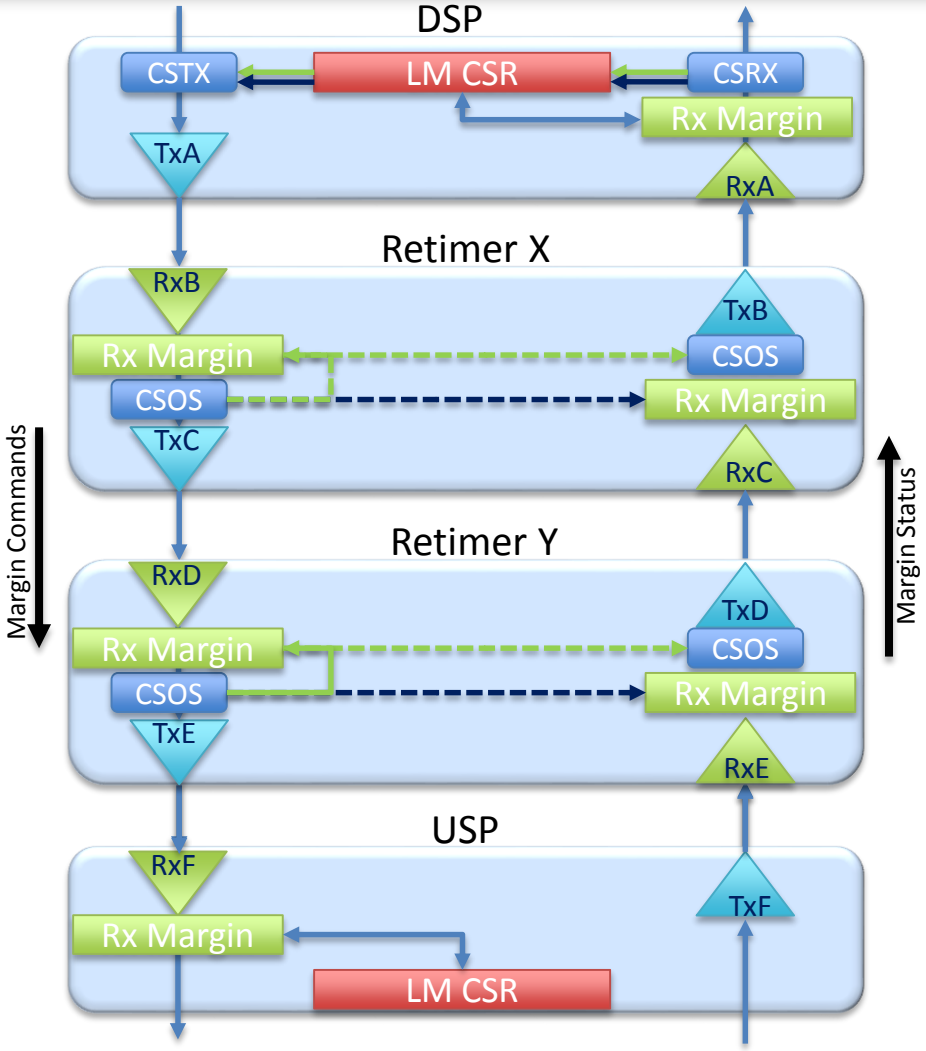
LM Blocks Retimer



Copyright © 2017 PCI-SIG® - All Rights Reserved

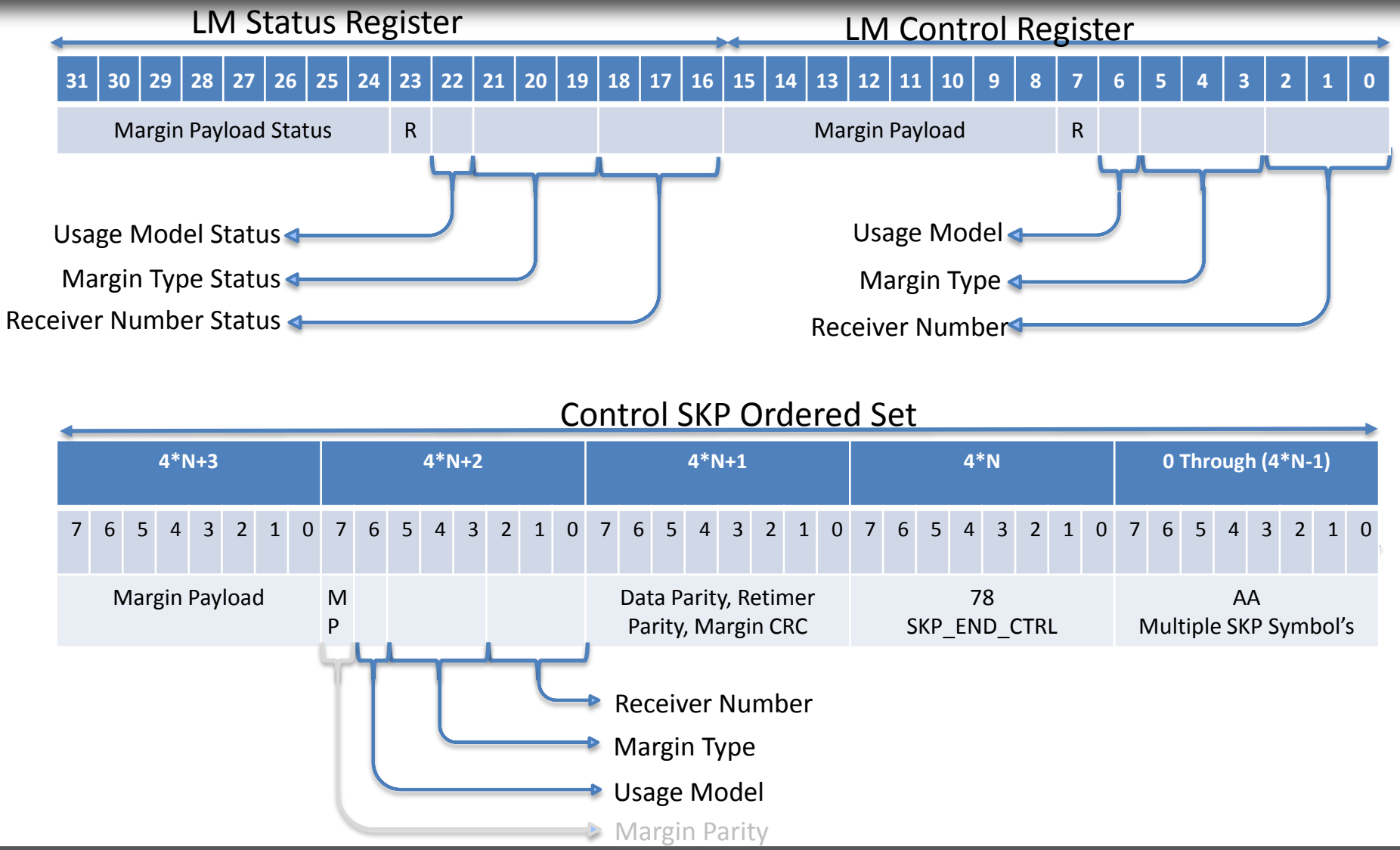
LM Topology

- **RxA**
 - Is margined using the DSP LM CSR
 - This is Controlled by DSP
- **RxB/RxC/RxD/RxE**
 - Margin Control is done by DSP using the Ctrl SKP OS sent to Retimers
 - Margin Status is transmitted back by Retimers to DSP using the Ctrl SKP OS
- **RxF**
 - Is margined using the USP LM CSR
 - This is Controlled by DSP by doing CFG Writes to USP LM CSR



Copyright © 2017 PCI-SIG® - All Rights Reserved

LM CSR/Ctrl SKP



Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Margin Commands

- **(Margin Type = 111) No Command**
 - Independent command used by DSP/USP as command and status based on certain conditions
- **(Margin Type = 001) Report Margin Commands**
 - Used to read a retimer register
 - Used to read the receiver capabilities such as supported offset, sampling method, etc
- **(Margin Type = 010) Set Commands**
 - Used to set error count limit
 - Used to move to normal settings
 - Used to clear error log.
- **(Margin Type = 011) Step Margin Timing Command**
 - Used to initiate lane margining at receiver for timing
- **(Margin Type = 100) Step Margin Voltage Command**
 - Used to initiate lane margining at receiver for voltage
- **(Margin Type = 101) Vendor Defined Command**
 - Used for vendor specific commands

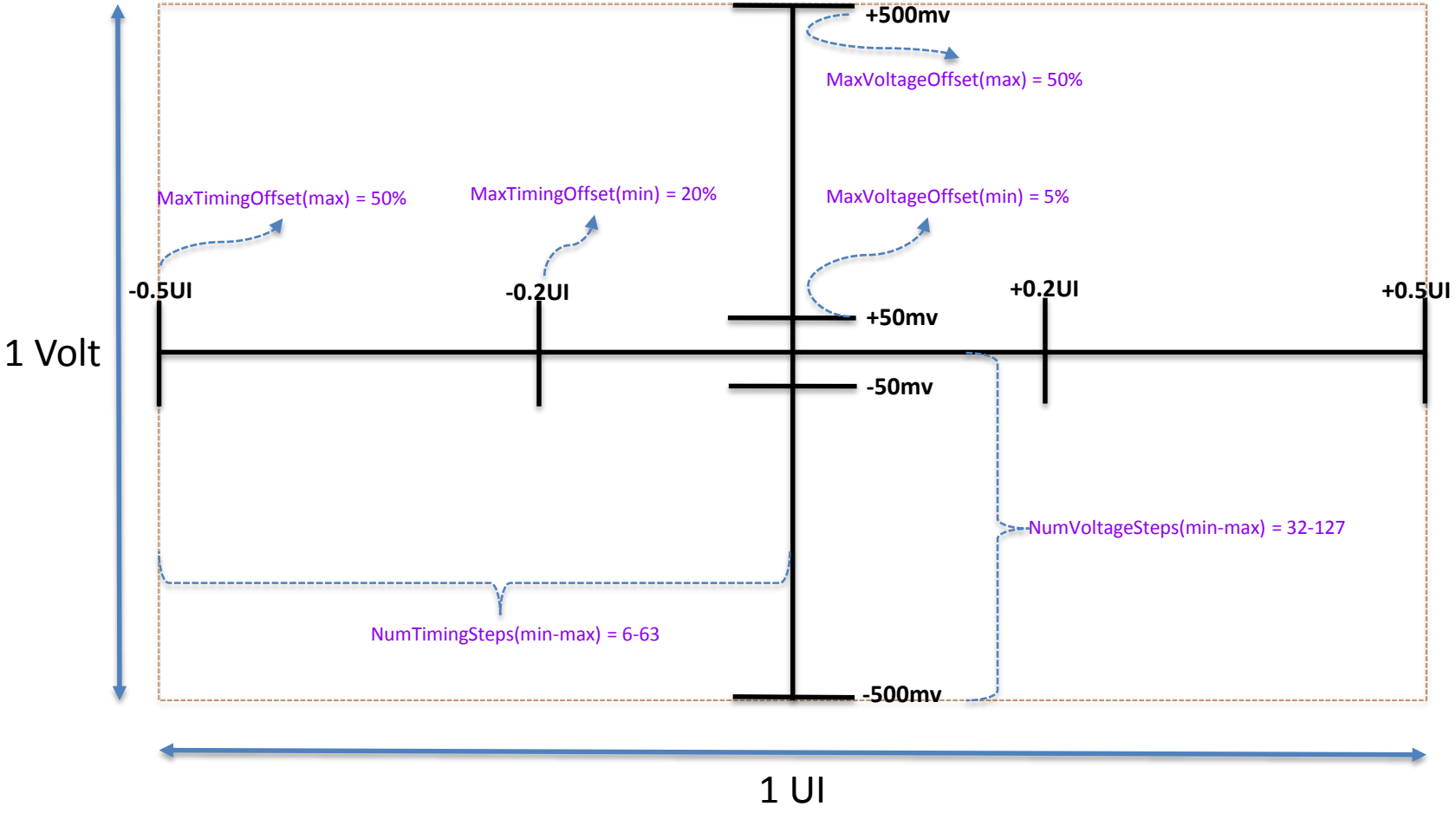
Copyright © 2017 PCI-SIG® - All Rights Reserved

PIPE 4.4.1 I/F

- **Adds per lane Message Bus Interface**
 - Time Division Multiplexed bus, that provides a way to initiate and respond to non-latency sensitive PIPE operations using minimal wires
 - M2P_Message Bus[7:0]
 - Used to Read/Write registers present in PHY
 - Used to send ACK response to P2M_Message received
 - P2M_Message Bus[7:0]
 - Used to Read/Write registers present in MAC
 - Used to send ACK response to M2P_Message received
- **Defines PHY & MAC Registers**
 - PHY implements “Control” Registers for holding the commands from MAC
 - MAC implements “Status” Registers for holding the status from PHY

Copyright © 2017 PCI-SIG® - All Rights Reserved

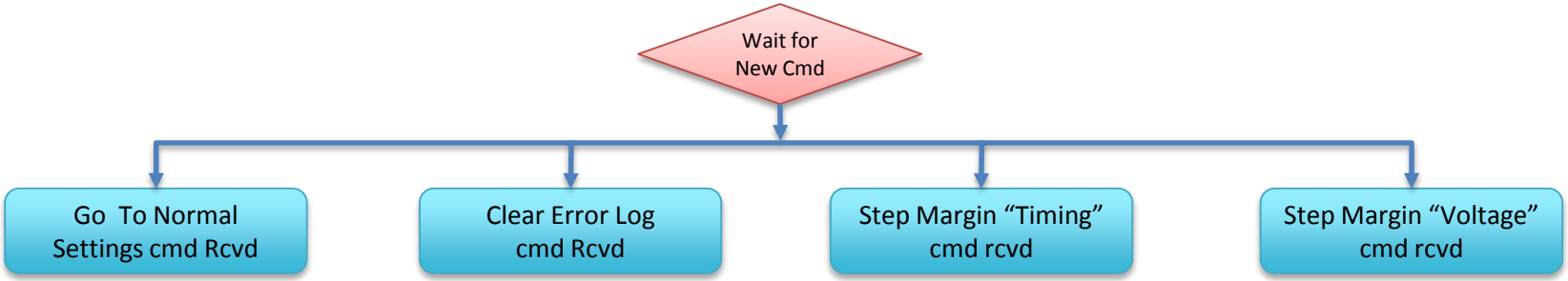
Margin Ranges



Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Commands←→ PIPE Commands

- **The Following LM commands get mapped to the PHY commands**

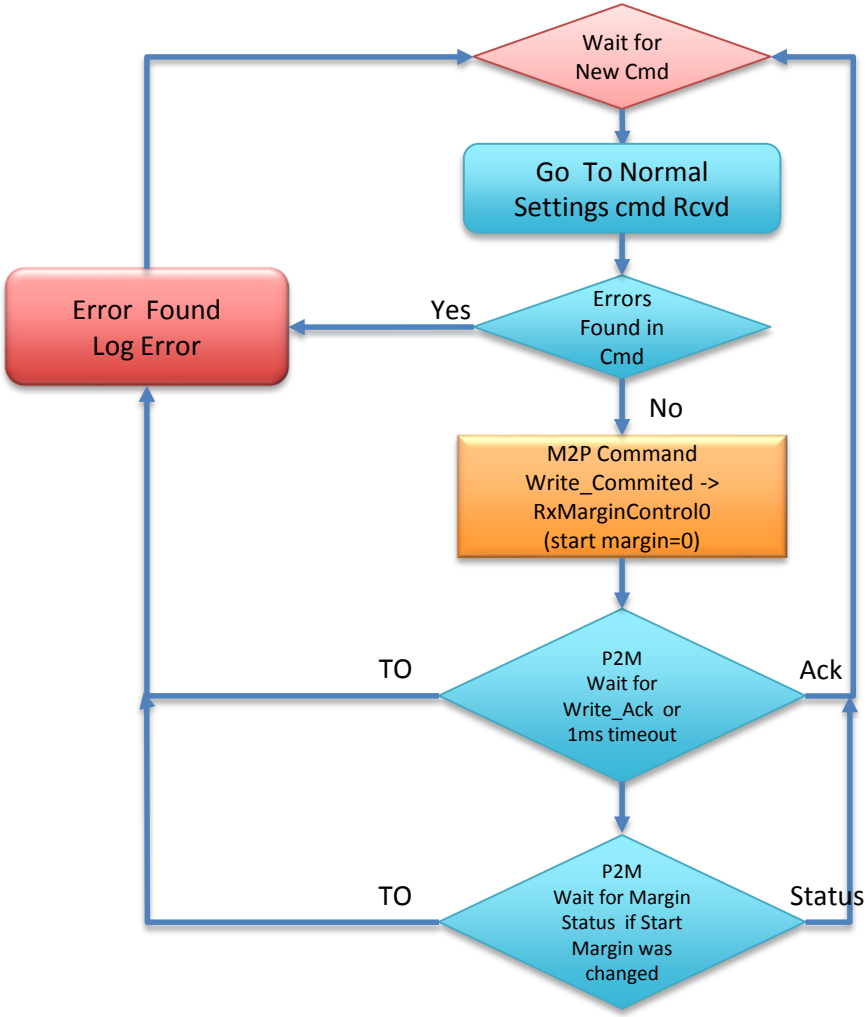


- **Other LM commands don't reach the PIPE I/F**
 - The following commands are terminated at MAC
 - No Command
 - Report Margin Parameters
 - Set Error Count Limit

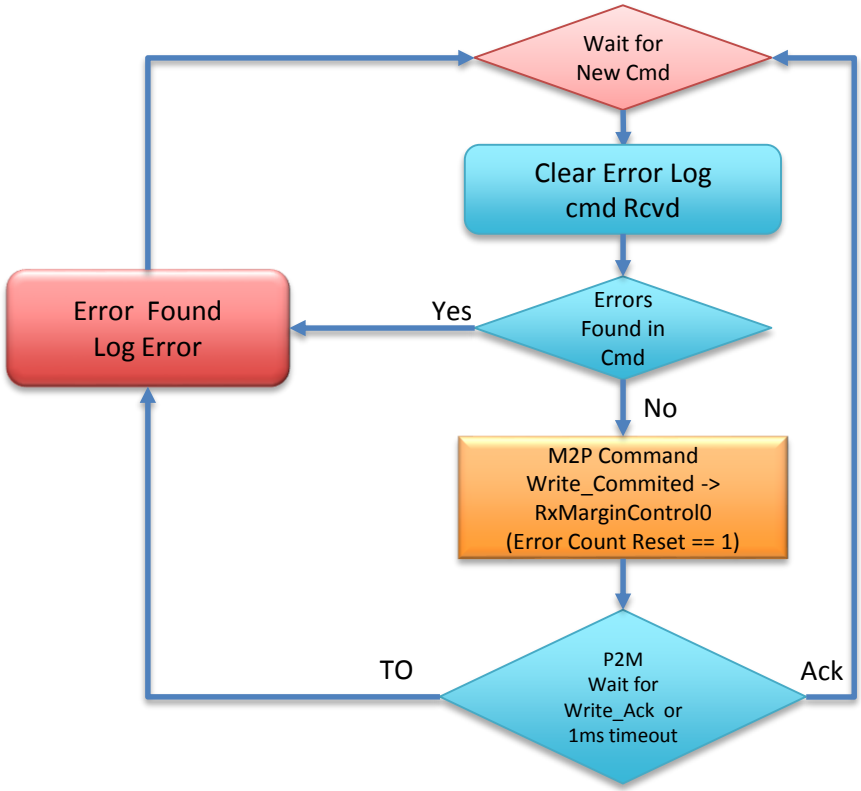
Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Commands←→ PIPE Commands

Go To Normal Command

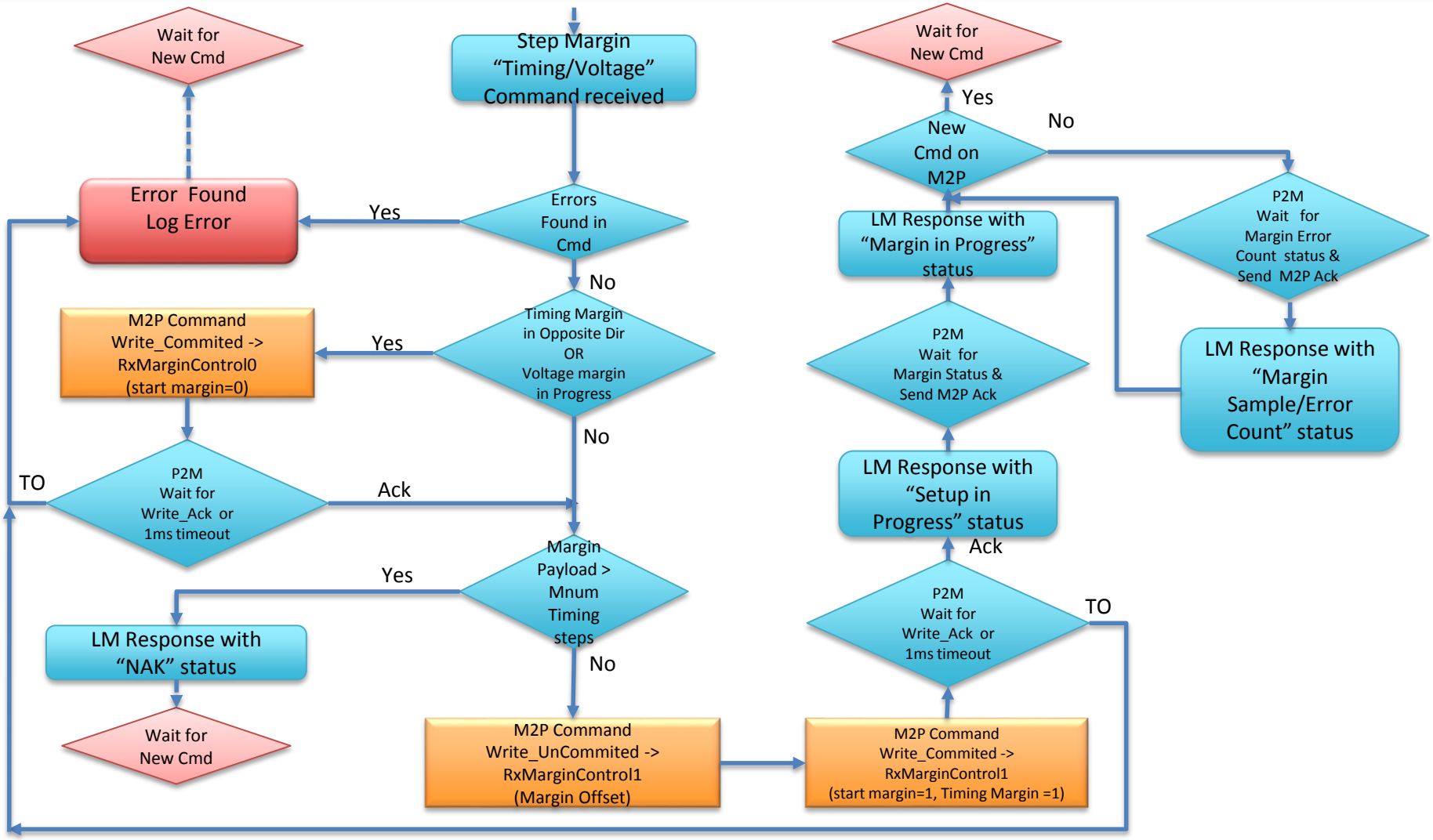


Clear Error Log Command



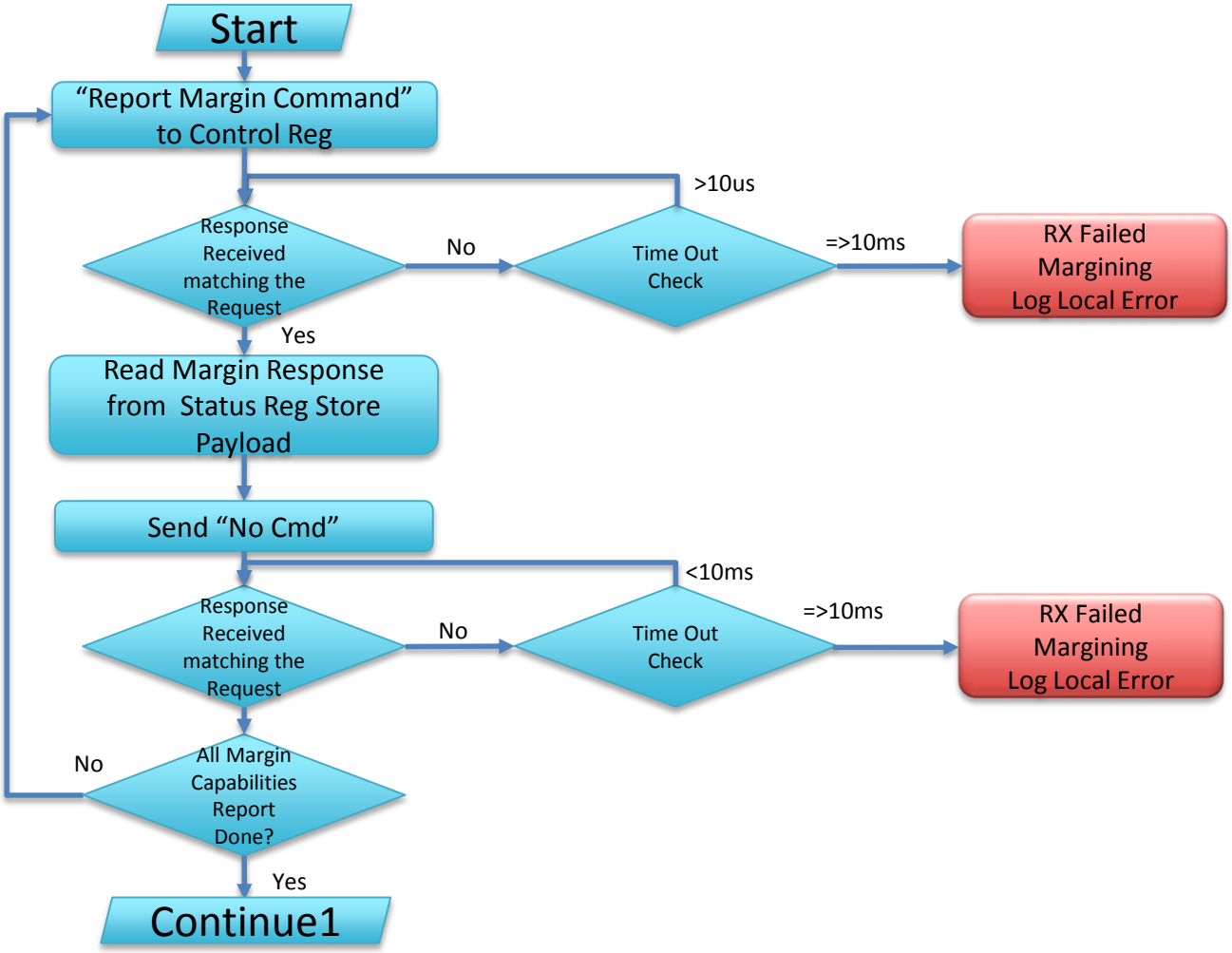
Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Step Margin←→ PIPE Commands



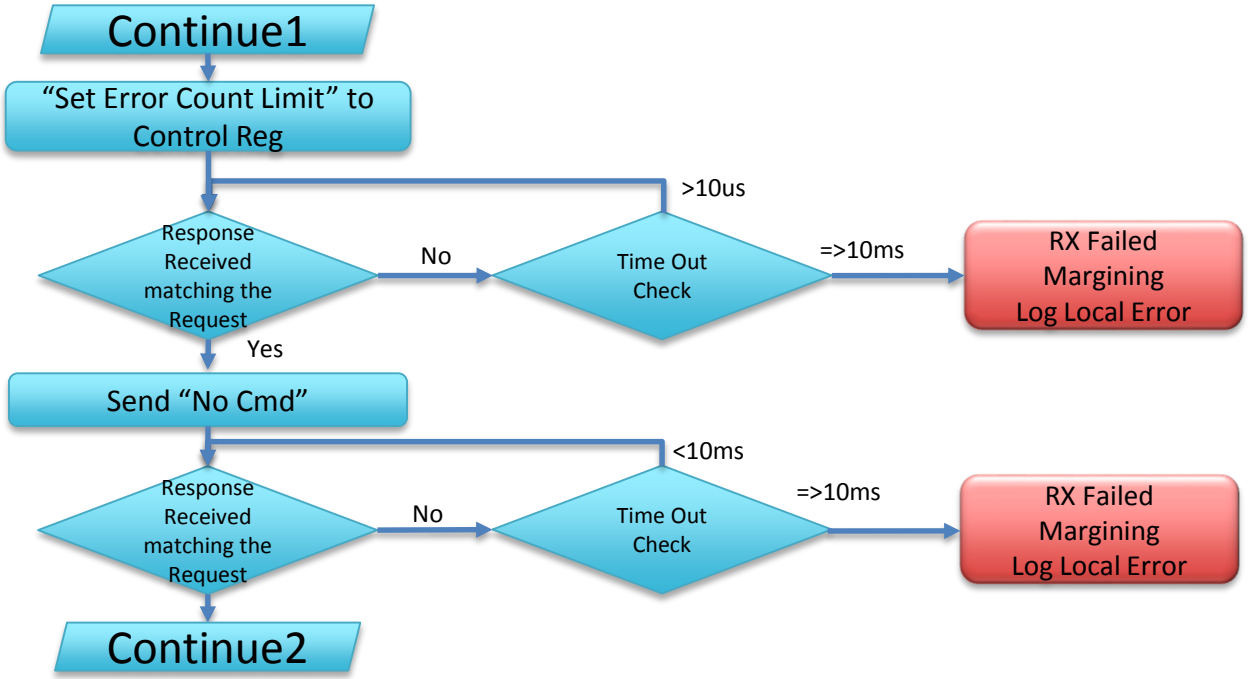
Copyright © 2017 PCI-SIG® - All Rights Reserved

LM S/W Flow – Report Cap



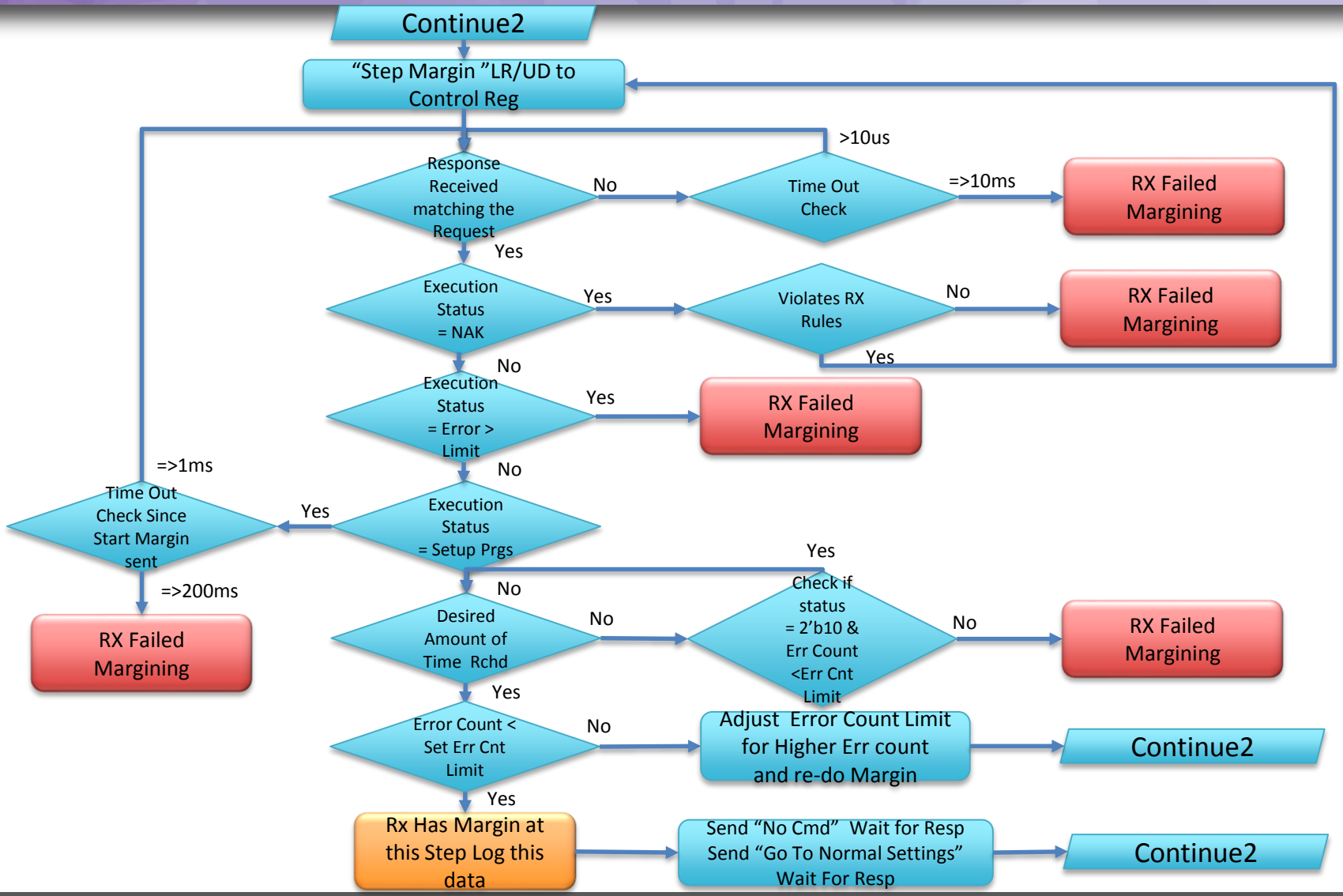
Copyright © 2017 PCI-SIG® - All Rights Reserved

LM S/W Flow – Set Err Cnt



Copyright © 2017 PCI-SIG® - All Rights Reserved

LM S/W Flow – Step Margin



Copyright © 2017 PCI-SIG® - All Rights Reserved

Sample/Error Count

- **As per Spec, at least10^8 bits margined before switching to next margin step location**
- **Typically PHY (PMA) will not implement 10^8 bits in a single iteration**
 - Programmable 4K to 250K bits per iteration
 - Set to 250K per sample in our design
- **Hence a single margining request from MAC could be split up into multiple iterations**
 - Accumulating the sample count over a period of time

Copyright © 2017 PCI-SIG® - All Rights Reserved

Sample/Error Count

- **M Sample Count = 3 * log2 (N)**
 - Where N is the number of actual bits tested during lane margining

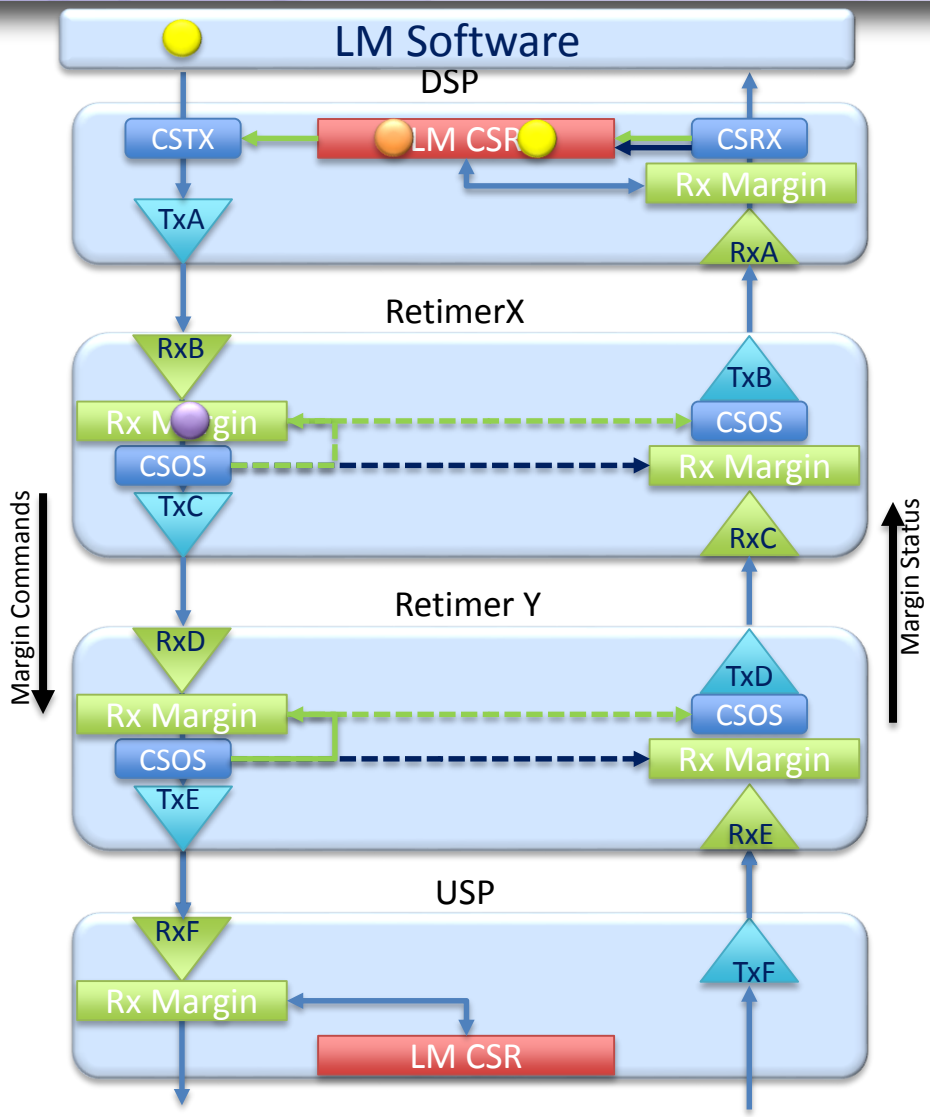
N Bits (No of bits Margined)	Iteration Counter Width (Implementation Specific)
4,096	30
8,192	29
16,384	28
32,768	27
65,536	26
131,072	25
262,144	24

- **Error Count**
 - Saturates if >= 63 errors are found.

Copyright © 2017 PCI-SIG® - All Rights Reserved

Flow(Retimer)

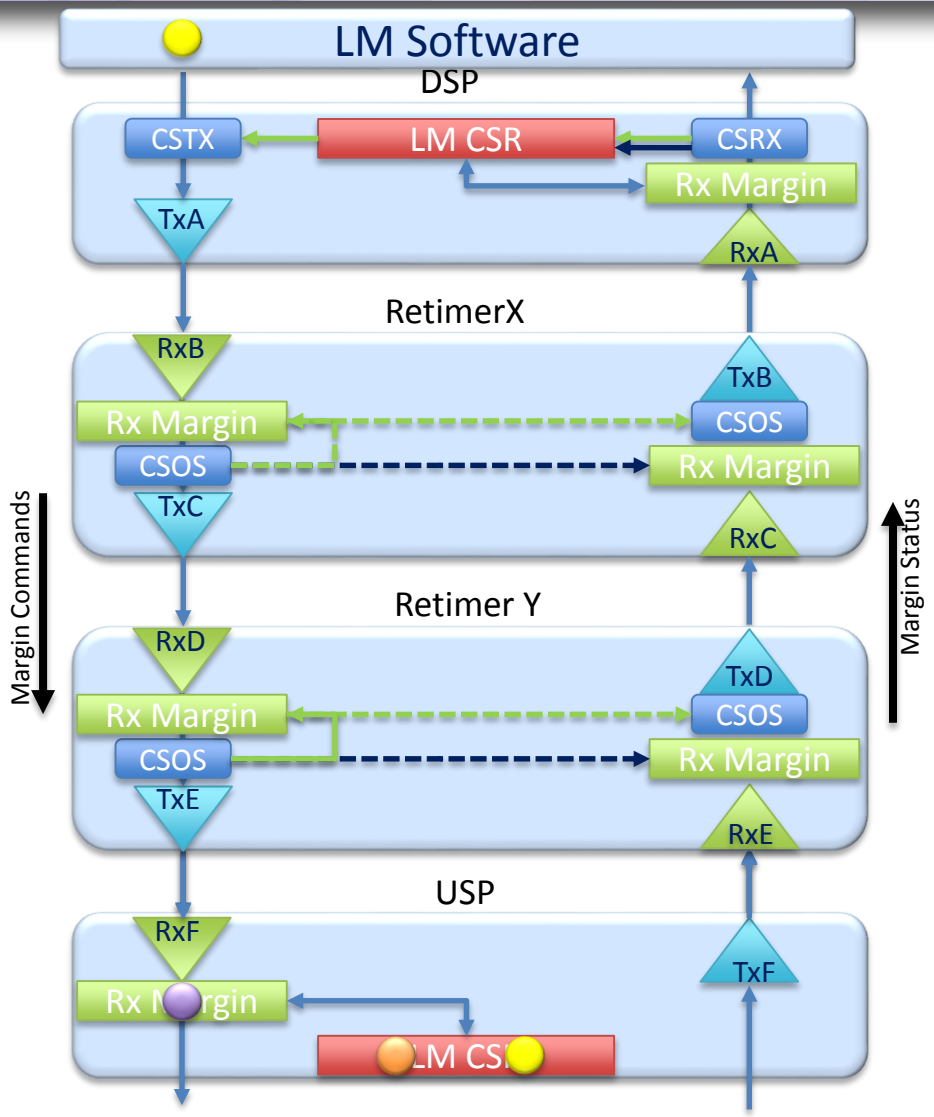
- **Command Flow from DSP to RxB**
- **Response from RxB to DSP**



Copyright © 2017 PCI-SIG® - All Rights Reserved

Flow(USP)

- **Command Flow from DSP to RxF in USP**
- **Response from RxF to USP Which DSP reads**



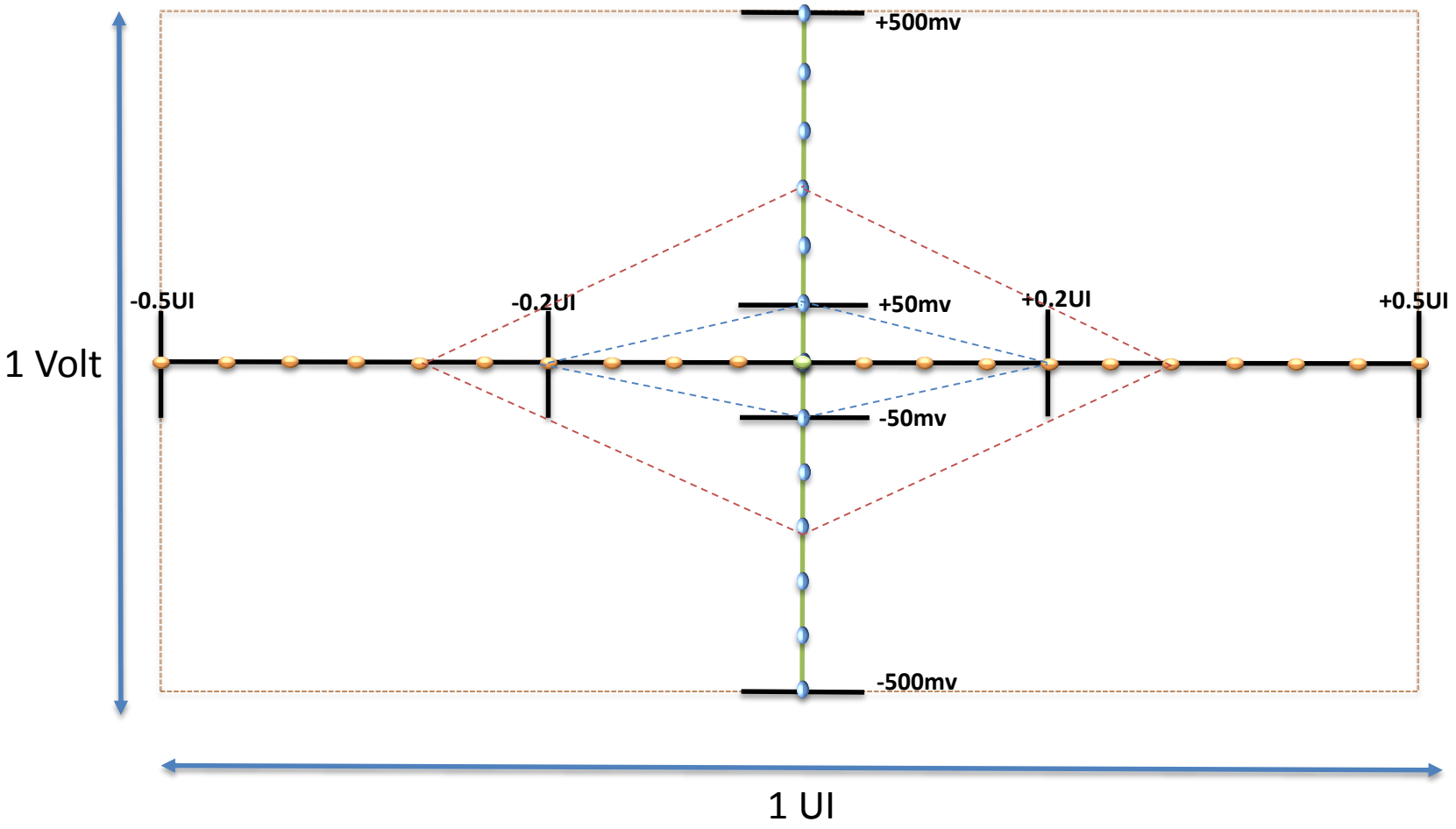
Copyright © 2017 PCI-SIG® - All Rights Reserved

Eye Plot Coordinates

- **Margining is done in 1D coordinates**
 - When stepping on Timing Axis
 - Voltage offset is kept ZERO
 - Timing Offset is swept and error measured
 - When stepping on Voltage Axis
 - Timing offset is kept ZERO
 - Voltage Offset is swept and error measured
- **1D measurement is sufficient to gather margin information**
 - Full Eye Plot or 2D scan of coordinates is not needed
 - 1D measurement is relatively simpler to implement and completes faster

Copyright © 2017 PCI-SIG® - All Rights Reserved

Eye Plot Coordinates (1D)



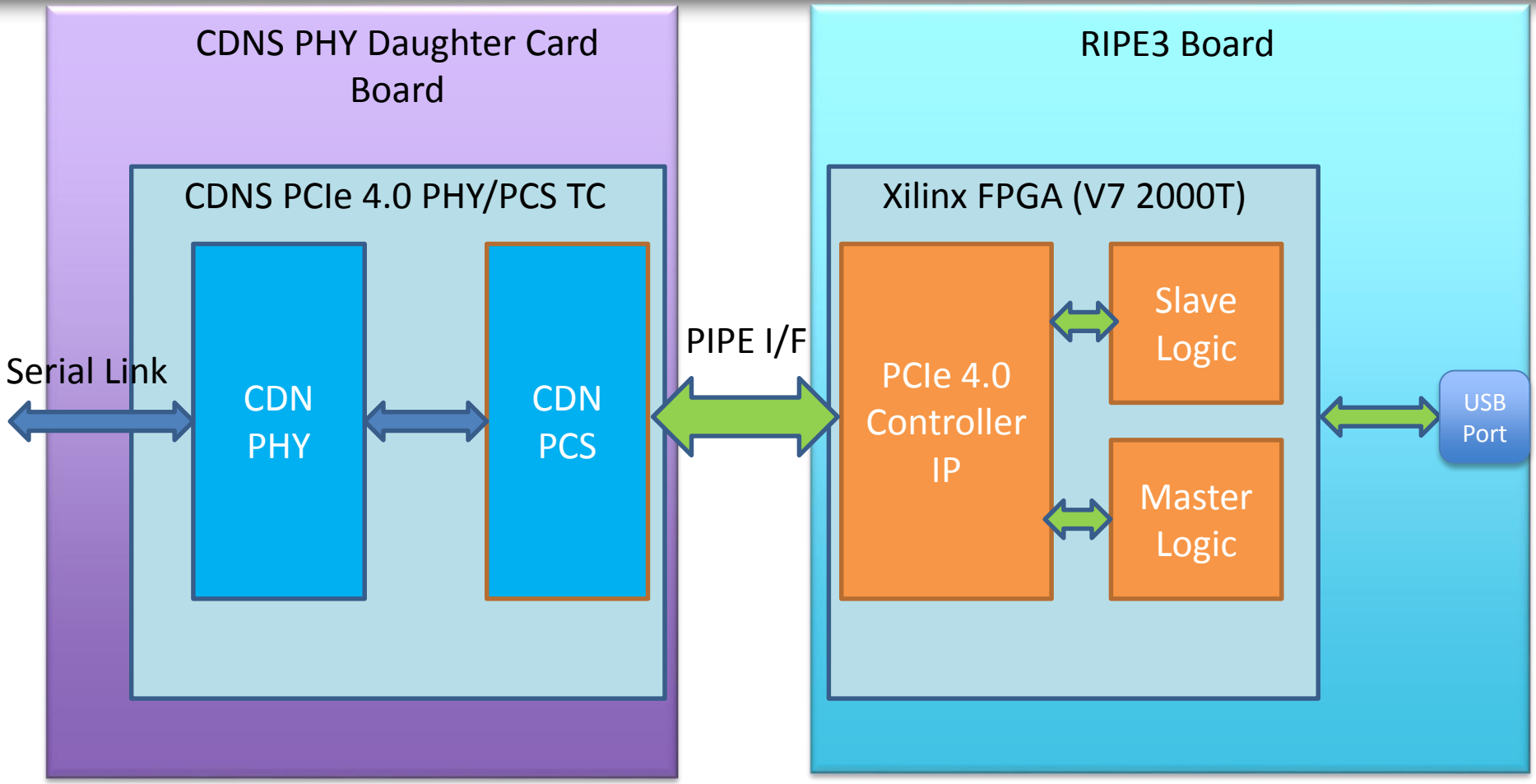
Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Validation

- **Validation Platform**
 - Controller+Phy Board
 - Controller is in the FPGA
 - PHY/PCS is a test Chip silicon
 - Connected at the PIPE I/F
 - Two Back 2 Back setups are used
 - Only has USP & DSP
- **S/W controlling the DSP via the USB port.**
 - Internally developed S/W
 - Plots eye, by sweeping both Voltage & Time (2D)

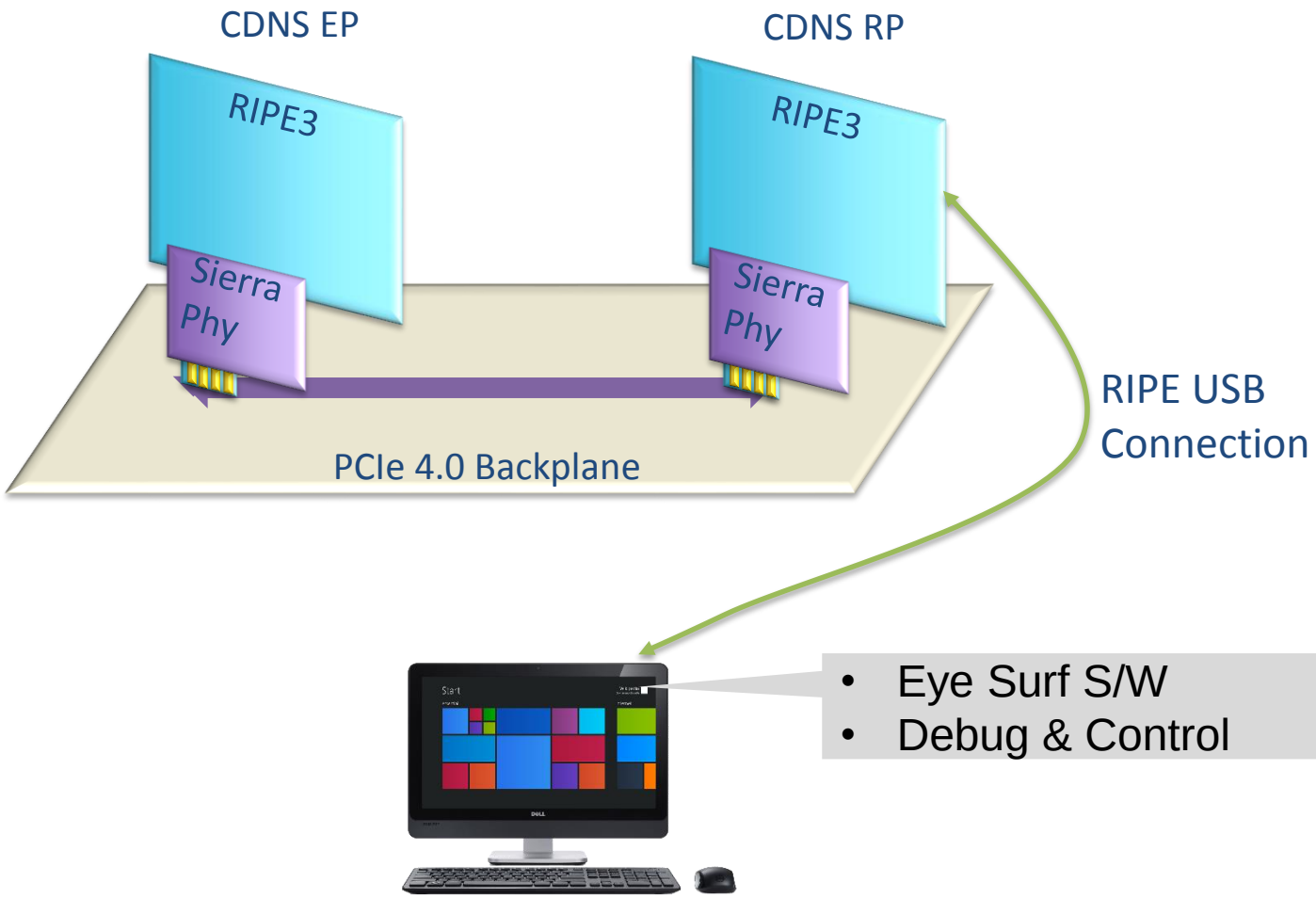
Copyright © 2017 PCI-SIG® - All Rights Reserved

Validation Platform Board



Copyright © 2017 PCI-SIG® - All Rights Reserved

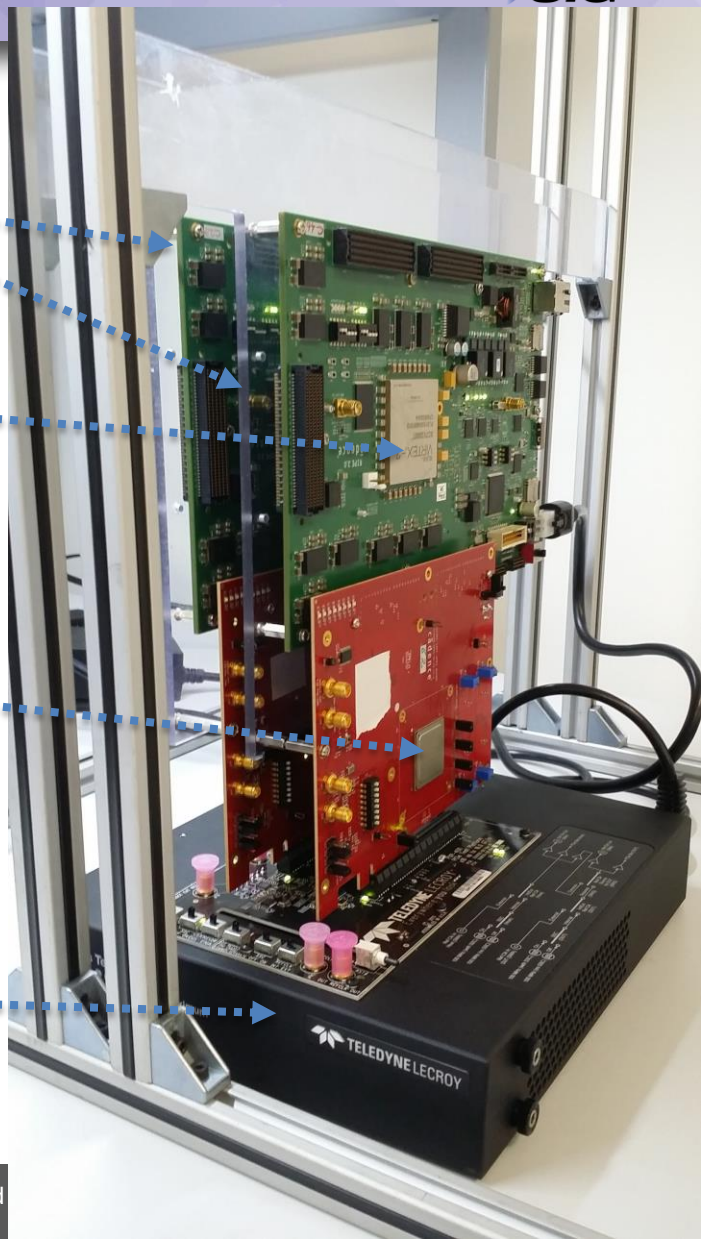
Validation Platform B2B



Copyright © 2017 PCI-SIG® - All Rights Reserved

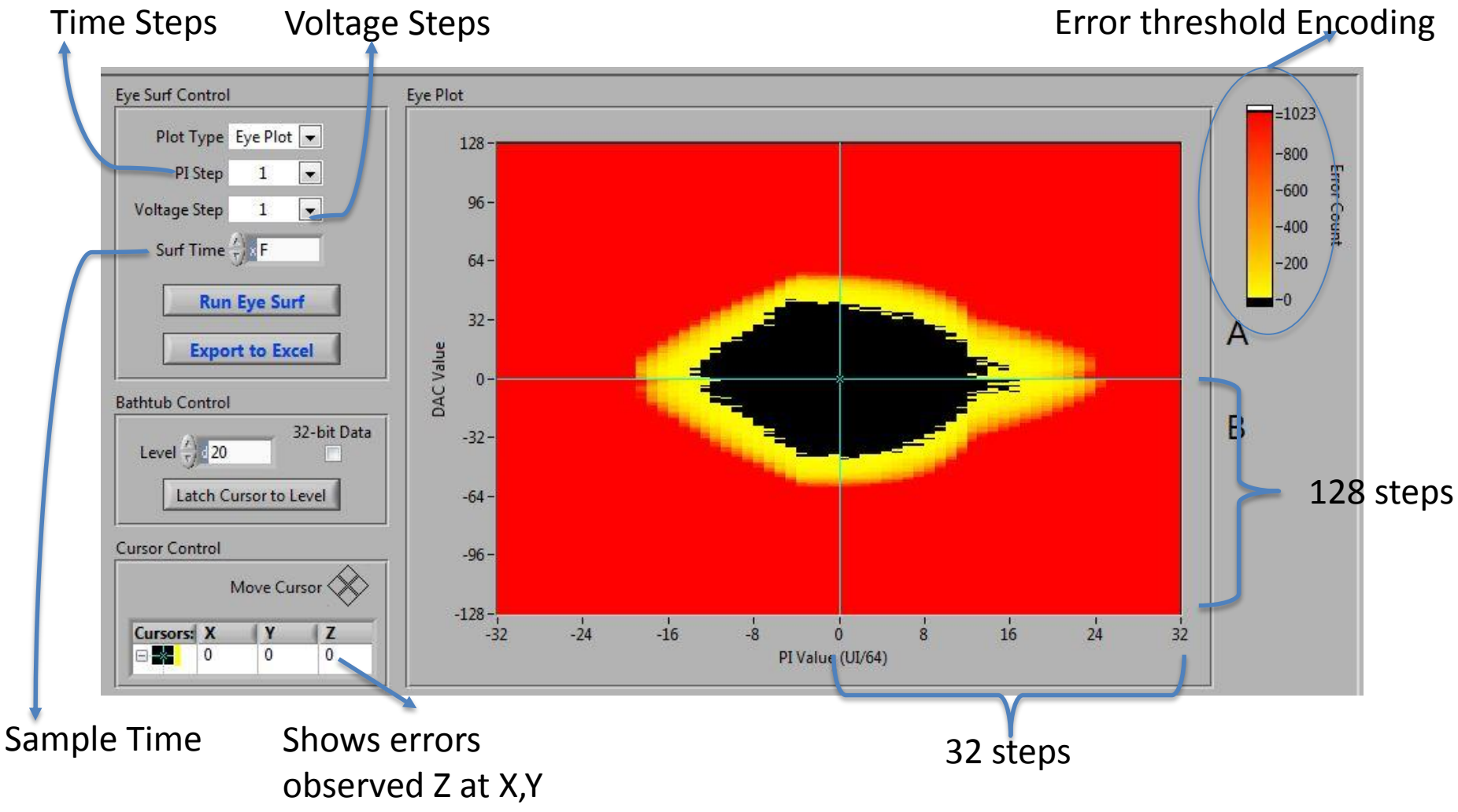
PCIe 4.0 B2B

- **RIPE3 Development board**
- **Cadence Controller**
 - RIPE3 Card FPGA Virtex7(-2)
- **Cadence PHY**
 - Sierra2 PHY
 - 32 bit PIPE @ 500MHz for PCIe 4.0
- **PCIe 4.0 Backplane**



Copyright © 2017 PCI-SIG® - All Rights Reserved

Example LM Plot (2D)



Copyright © 2017 PCI-SIG® - All Rights Reserved

Error Scenarios- S/W & PIPE

- **S/W Error Scenarios**
 - All Error scenarios with S/W taken care by timeout mechanism
 - S/W switching between Voltage to Timing without a NO Command & Go To Normal
 - Logic to gracefully terminate PIPE I/F Voltage Margin command that is already in progress
 - Reserved commands received will be dropped/ignored
 - S/W will see timeout waiting for response
- **PIPE I/F Error Scenarios**
 - Response from PHY/MAC not received
 - Logic to timeout and log error and then wait for next legal command

Copyright © 2017 PCI-SIG® - All Rights Reserved

Error Scenarios – P2M/M2P

- **Commands with illegal length**
 - Commands not following 3/2/1 cycle format. This may lead to next command being misinterpreted.
 - Illegal command is dropped
 - Logic will wait for a NOP and then resume command decoding
- **Invalid Write Address/Write Attributes**
 - No defined way to inform the PHY/MAC about this
 - These commands are dropped, if a response is expected Time Out mechanism will detect this error
- **Simultaneous Write Committed from PHY & MAC**
 - Logic should handle generating Write ACK without going into a deadlock

Copyright © 2017 PCI-SIG® - All Rights Reserved

Error Scenarios – P2M/M2P

- **A new "Start Margin" command initiated illegally**
 - MAC logic is responsible to stop the previous margin if already in progress before issuing new margin command
- **MAC moving PHY out of P0 or 16GT/s while margining is in progress**
 - Both MAC & PHY should terminate their respective LM state machine and return to Idle
- **PHY looses data lock when in 16GT/s (Independent Error Sample case)**
 - Both PHY & MAC should reset their LM State machines
 - All pending commands may not be responded and will cause time outs

Copyright © 2017 PCI-SIG® - All Rights Reserved

Summary

- **LM is mandatory for 16GT/s and greater speeds**
- **LM defines standard way to margin all RX in the link**
- **Changes needed in PCIe 4.0 Controller, PIPE I/F & PCS/PMA to support this**
- **S/W driver helps in capture data and plot**
- **Design and test LM Error Scenarios**

Copyright © 2017 PCI-SIG® - All Rights Reserved

References

- **PCIe 4.0 ver. 0.7 Spec**
 - Link to Base 4.0 rev 0.7 (Nov 11, 2016)
- **PIPE 4.4.1 Spec**
 - Link to PIPE 4.4.1 Spec

Copyright © 2017 PCI-SIG® - All Rights Reserved

Acknowledgement

- **Would like to thank following Cadence PCIe team members for helping with this presentation**
 - Chetan Paragaonkar
 - Fred Stivers
 - Gary Dick
 - Lana Chan
 - Sachin Dhingra
 - Sathish Kumar Ganesan

Copyright © 2017 PCI-SIG® - All Rights Reserved

Thank you for attending the PCI-SIG Developers Conference 2017.

For more information please go to www.pcisig.com

Copyright © 2017 PCI-SIG® - All Rights Reserved



Backup

Copyright © 2017 PCI-SIG® - All Rights Reserved

What is LM?

- **Spec defined new Capability Structure**
 - Present only in USP & DSP
- **Two error sampling Architectures**
 - Independent Error Sampler
 - Error is measured on the copy of original data received
 - Margining will not produce errors in Data (*)
 - This presentation talks about this method only.
 - Actual Data Sampler
 - Error is measured on the original data received
 - Margining “may” produce errors in Data

Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Basic Rules

- **Always initiated by Downstream Port (Master)**
 - Upstream Ports and Retimers Respond to this request (Slaves)
 - DSP RX are margined through its own CSR
 - USP RX are margined via its own CSR, controlled by DSP Write/Read to CSR in USP
 - Retimers RX are Margined via Control SKP OS, controlled by DSP CSR
- **New PIPE 4.4.1 I/F**
 - MAC & PHY communicate the Lane Margining command and status via
 - MAC-to-PHY(M2P) message bus
 - PHY-to-MAC(P2M) message bus

Copyright © 2017 PCI-SIG® - All Rights Reserved

Ctrl SKP OS Handling

- **DSP**
 - TX Ctrl SKP OS contains the current command in the Lane Control Reg
 - Rx Ctrl SKP OS update the status only if Retimer is present in the link else its dropped
- **USP**
 - Received Ctrl SKP OS are ignored/dropped by the EP
 - Transmitted Ctrl SKP OS are sent with NO command
- **Retimers**
 - Received Ctrl SKP OS is accepted if the Rx Number Matches else its dropped
 - Transmitted Ctrl SKP OS are sent with response to Command (continuously), else sends NO command.

Copyright © 2017 PCI-SIG® - All Rights Reserved

PIPE 4.4.1 Regs - MAC

- **Located in MAC that PHY will access**

Address	Reg Name	Description
12'h0	RX Margin Status0	**Margin Nak**: PHY Indicates the received margin command is invalid **Margin Status** : Phy Acknowledges a valid change to "start margin" request from MAC
12'h1	RX Margin Status1	**Sample Count**: Indicates number of bits that have been margined.
12'h2	RX Margin Status2	**Error Count**: Valid for "Ind Error Sample" , indicates the actual bit errors seen by PHY.
12'h3	Elastic Buffer Status	**Elastic Buffer Status**: Bit Indicates PHY has adjusted the Elastic Buffer depth to the value indicated in the "Elastic Buffer Ctrl Reg"
Reserved	NA	Reserved

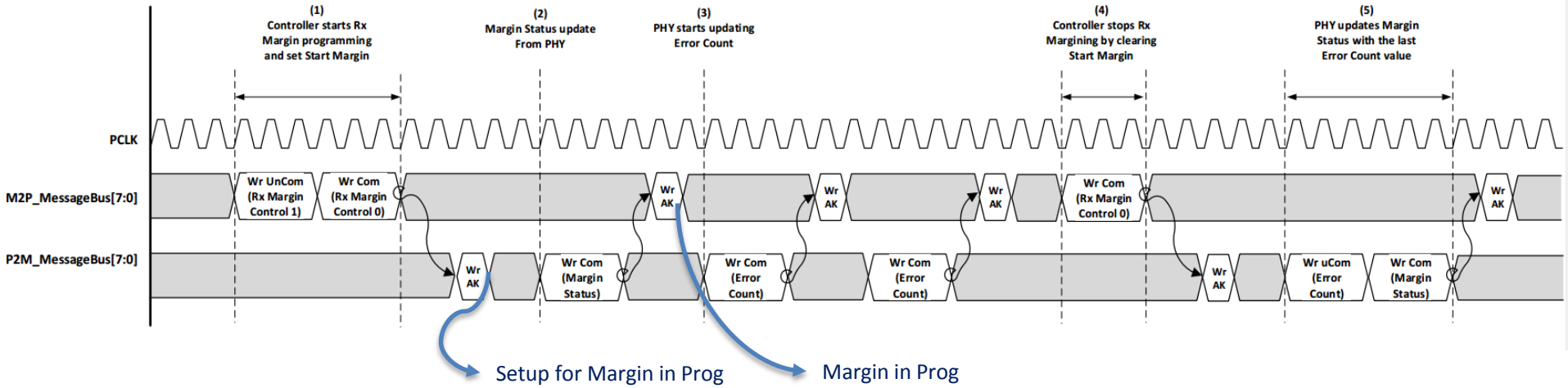
Copyright © 2017 PCI-SIG® - All Rights Reserved

PIPE 4.4.1 Regs - PHY

- **Located in PHY that MAC will access**

Address	Reg Name	Description
12'h0	RX Margin Control0	**Sample Count Reset**: Used to reset the "Sample Count" in RX Margin Status1 **Error Count Reset**: Used to reset the "Error Count" in Rx Margin Status2 **Margin Voltage/Timing**: 0 Selects Voltage Margining, 1 Selects Timing Margining **Start Margin**: Used to Start and Stop Margining
12'h1	RX Margin Control1	**Margin Direction**: 0 Steps timing Left or Voltage Up, 1 Steps Timing Right or Voltage Down **Margin Offset**:Margin Offset from default position
12'h2	Elastic Buffer Control	**Elastic Buffer Control**: Used to set the elastic Buffer Depth. Beneficial for Retimers to increase depth when SRIS mode is enabled.
Reserved	NA	Reserved

PIPE 4.4.1 Start/Stop Margining



(1) Margin Command received by MAC

(2) Margin Status updated by PHY

(3) PHY Updates Error Count "Continuously"

(4) MAC stops RX Margining

(5) PHY Updates final values of Status & Error Count

Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Topology

- **Command/Status Flows**
 - Margin Commands flows downstream
 - Margin Status flows upstream
- **RP/EP has Margin Control Status Register (CSR)**
- **Retimers don’t have CSR**
 - Receive control information via RX Control SKP OS
 - Transmit status information via TX Control SKP OS
- **For each received margin command**
 - The receiver that was targeted has to respond with appropriate margin status

Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Port Cap/Status

- **Margining Port Capabilities**
 - **Margining uses Driver Software** [0]
 - If Set, indicates that Margining is partially implemented using Device Driver software. “Margining Software Ready” indicates when this software is initialized
 - If Clear, Margining does not require device driver software. In this case the value read from “Margining Software Ready” is undefined
- **Margining Port Status**
 - **Margining Ready[0]**
 - Indicates when the Margining feature is ready to accept margining commands
 - Indicates in 16GT/s & L0
 - **Margining Software Ready[1]**
 - When “Margining uses Driver Software” is Set, then this bit, when Set, indicates that the required software has performed the required initialization

Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Command Fields

- **Usage Model**
 - 0b indicates that the usage model is Lane Margining at Receiver.
 - 1b in this field is reserved for future usages
- **Receiver Numbers**
 - Targets different receivers, details on next slide
- **Margin Command**
 - Margin command is an combination of {Margin type, Receiver number, Margin Payload}
 - Based on the margin command, the selected receiver will accept the command
 - For the accepted command the selected receiver has to respond with appropriate margin status

Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Rx Numbers

- **From DSP**
 - RX Number Encodings are used by the DSP for Margin Commands targeting that DSP/Retimer below that Downstream Port
- **From USP**
 - RX Number Encodings are used by the USP for Margin Commands targeting that Upstream Port

RX Number	From DSP	From USP
000b	RxA…..RxE (Broadcast)	RxF (Broadcast)
001b	RxA	Reserved
010b	RxB	Reserved
011b	RxC	Reserved
100b	RxD	Reserved
101b	RxE	Reserved
110b	Reserved	RxF
111b	Reserved	Reserved

Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Margin Commands/Responses

Command Name	Type	RX Num	Payload	Type	RX Num	Payload
No Command	111b	000b	9ch	111b	000b	9ch
Access Retimer register	001b	010b, 100b	Reg Offset 00h-87h, A0h-FFh	001b	010b, 100b	Reg Value if supported else 00h
Report Margin Parameters	001b	001b to 110b	88h to 90h	001b	001b to 110b	Response for the request
Set Error Count Limit	010b	001b to 110b	Margin Payload {2'b11,Error Count Limit}	010b	001b to 110b	Margin Payload {2'b11,Error Count Limit Registered by Rx}
Go To Normal Settings or Clear Error Log	010b	000b to 110b	0Fh or 55h	010b	001b to 110b	0Fh or 55h

Copyright © 2017 PCI-SIG® - All Rights Reserved

LM Margin Commands/Responses

Command Name	Type	RX Num	Payload	Type	RX Num	Payload
Step Margin Timing (Left/Right)	011b	001b to 110b	Margin Amount Payload[7] = Reserved Payload[6] = 0 Right , 1 Left Payload[5:0]: No Of T Steps	011b	001b to 110b	{Step Margin Execution Status, MErrorCount} Execution Status as follows 11: Nak 10: Margin in Prog 01: Setup for Margin in Prog 00: Error > Limit
Step Margin Voltage (Up/Down)	100b	001b to 110b	Margin Amount Payload[7] = Reserved Payload[6] = 0 Up , 1 Down Payload[5:0]: No Of V Steps	100b	001b to 110b	{Step Margin Execution Status, MErrorCount} Execution Status as follows 11: Nak 10: Margin in Prog 01: Setup for Margin in Prog 00: Error > Limit
Vendor Defined Command	101b	001b to 101b	Vendor Defined Payload	101b	001b to 101b	Vendor Defined Payload

Copyright © 2017 PCI-SIG® - All Rights Reserved

PIPE 4.4.1 I/F

- **Simple Request-Handshake implemented to access the registers**
 - Both MAC and PHY can initiate Message Bus Requests
 - And access each others Control/Status Registers
- **For each initiated Message Bus Request**
 - The receiver has to respond with appropriate Message Bus Handshake
 - This indicates the completeness of the initiated Request
- **Whenever PHY finishes one iteration of Rx Margining**
 - PHY will access the MAC Status registers to update the Status.
 - MAC will re-map the PHY status update into the Step Margin Execution Status for the system software

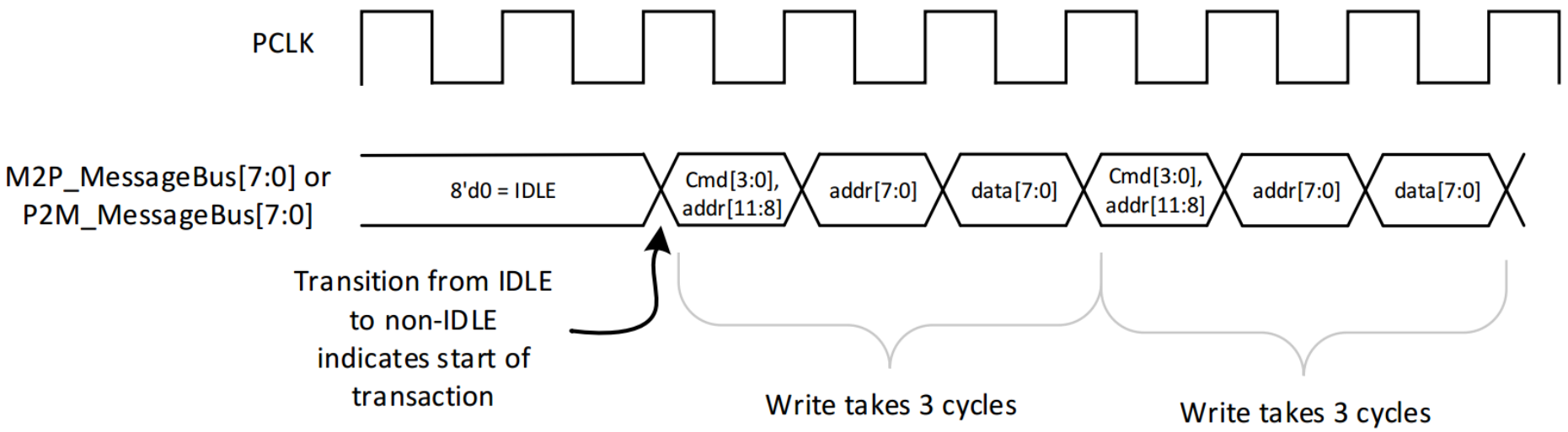
Copyright © 2017 PCI-SIG® - All Rights Reserved

PIPE 4.4.1 Commands

Command	Cmd Encoding	Description	Transmit Cycles
NOP	4'b0000	Idle Command	C0:{Cmd[3:0],4'd0}
Write Uncommitted	4'b0001	Current Write is saved off to a shadow register and only updated when Write Committed is received	C0:{Cmd[3:0],Addr[11:8]} C1:{Addr[7:0]} C2:{Data[7:0]}
Write Committed	4'b0010	Current write and previously received uncommitted writes are updated to final regs.	C0:{Cmd[3:0],Addr[11:8]} C1:{Addr[7:0]} C2:{Data[7:0]}
Read	4'b0011	Used to read a PIPE Register	C0:{Cmd[3:0],Addr[11:8]} C1:{Addr[7:0]}
Read Completion	4'b0100	Data Response to a Read Request.	C0:{Cmd[3:0],4'd0} C1:{Data[7:0]}
Write Ack	4'b0101	Acknowledge receipt of a Write Committed request.	C0:{Cmd[3:0],4'd0}
Reserved	All Others	N/A	N/A

Copyright © 2017 PCI-SIG – All Rights Reserved

PIPE 4.4.1 Commands



- **Message bus uses TDM where complete command sent out 8 bits at a time every clock cycle**
- **Request sent on M2P that expect a response will come on P2M and vice versa**

Copyright © 2017 PCI-SIG® - All Rights Reserved